Problem 4

This is a *stiffly oscillatory problem*, since it has two oscillatory modes, a slow one and a fast one, along with a slowly decaying mode, combined together:

$$y^{(0.5)} = \frac{1}{8}\begin{pmatrix} 41 & 41 & -38 & 40 & -2 \\ -79 & 81 & 2 & 0 & -2 \\ 20 & -60 & 20 & -20 & -8 \\ -22 & 58 & -24 & 20 & -4 \\ 1 & 1 & -2 & -4 & -2 \end{pmatrix} y, \qquad t \in [0, 20], \qquad y(0) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}.$$

Its solution is depicted in Figure 7. The width of the integration interval is chosen in order to appreciate both the two oscillatory modes, as well as the initial decay of the last mode.

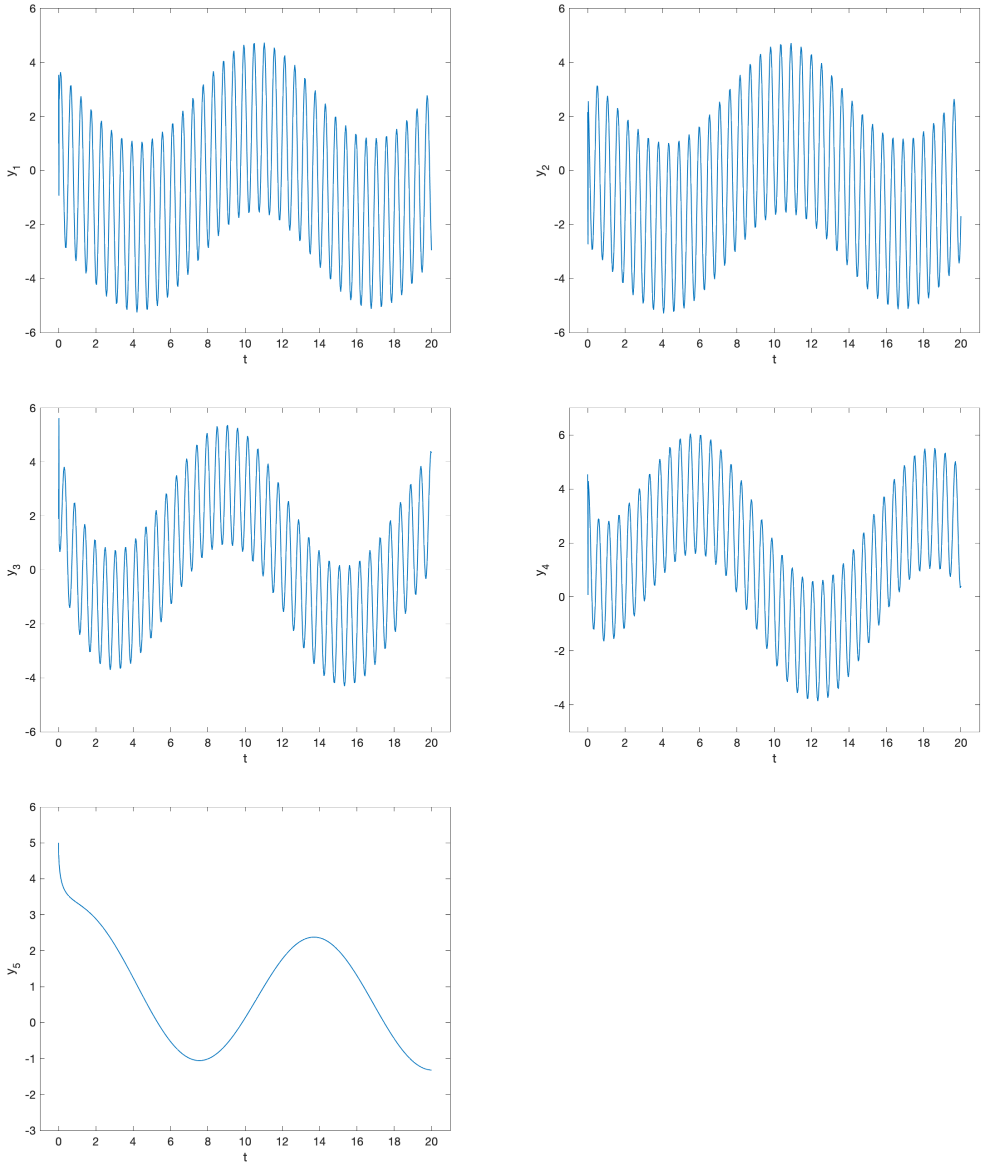

Figure 7. Components of the solution of Problem 4.

For this problem we cannot consider the codes `fcoll` and `tsfcoll`, which are only able to solve scalar problems. We use, therefore, the following codes, with the parameters chosen as follows:

- `fde12`, `fde12-10`: $\mathtt{h} = 10^{-i}$, $i = 5, 6$;

- `flmm2-1`, `flmm2-2`, `flmm2-3` : $\mathtt{h} = 5 \cdot 10^{-i}$, $i = 4, 5$;
- `fhbvm`: $\mathtt{M} = 100i$, $i = 3, 4, 5$;
- `fhbvm2`: $\mathtt{nu} = 50$, $\mathtt{n} = 1$, $\mathtt{N} = 100i$, $i = 3, 4, 5$.

The obtained results are summarized in Figure 8, from which one deduces that:

- `fde12-10` can reach an accuracy of 3 mescd, higher than that of `fde12`, but with a high execution time (more than 2500 sec);
- `flmm2-1` is more accurate than `flmm2-2`, which in turn is more accurate than `flmm2-3` (all methods reach about 2 mescd), and all requiring about 700 sec of execution time;
- the codes `fhbvm` and `fhbvm2` are able to reach more than 10 significant digits, in a much lower time. In particular, the double-mesh implementation used in the code `fhbvm2`, which is particularly efficient for solving problems with oscillatory solutions, results into an execution times of 1.5 sec, whereas `fhbvm` requires 70 sec (though it is slightly more accurate than `fhbvm2`).

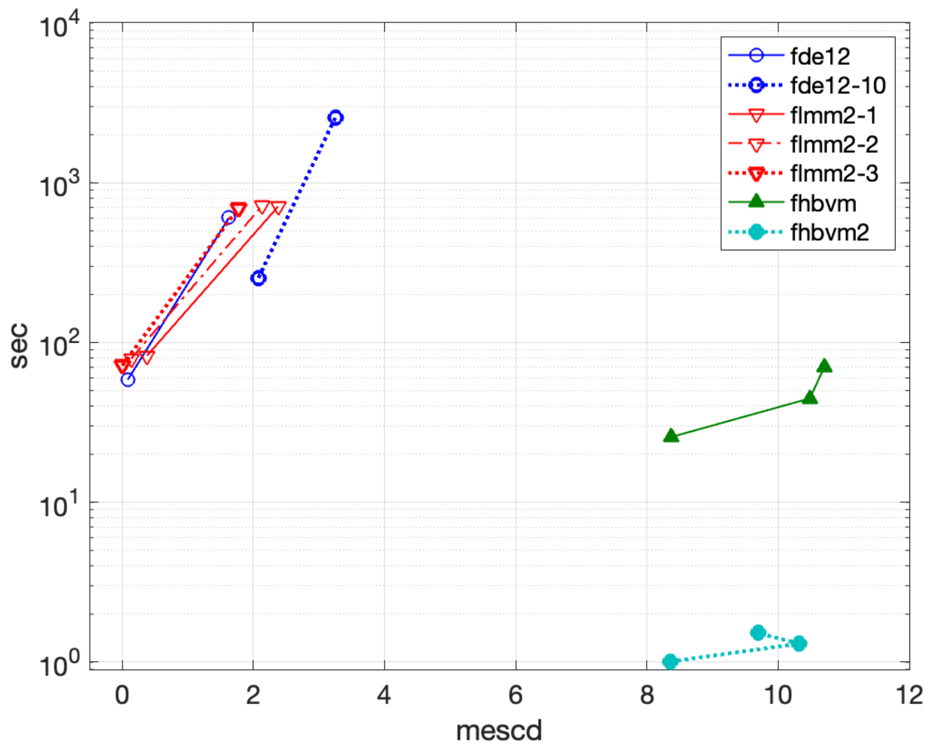

Figure 8. WPD for Problem 4.